6D Frictional Contact for Rigid Bodies

C. Bouchard
McGill University, Canada

M. Nesme
LJK Inria, France

M. Tournier
RIKEN BSI, BTCC, Japan

B. Wang
Shenzhen VisuCA Key Lab, SIAT, China

F. Faure
LJK Inria, UJF, France

P. G. Kry*
McGill University, Canada

ABSTRACT

We present a new approach to modeling contact between rigid objects that augments an individual Coulomb friction point-contact model with rolling and spinning friction constraints. Starting from the intersection volume, we compute a contact normal from the volume gradient. We compute a contact position from the first moment of the intersection volume, and approximate the extent of the contact patch from the second moment of the intersection volume. By incorporating knowledge of the contact patch into a point contact Coulomb friction formulation, we produce a 6D constraint that provides appropriate limits on torques to accommodate displacement of the center of pressure within the contact patch, while also providing a rotational torque due to dry friction to resist spinning. A collection of examples demonstrate the power and benefits of this simple formulation.

Index Terms: Computer Graphics [I.3.5]: Computational Geometry and Object Modeling—Physically based modeling

1 INTRODUCTION

Most contact models assume that the contact geometry is a point. While this is sufficient to compute repulsion and friction forces, it does not include enough information to produce resistance to rolling or spinning. The rotational behavior of contact only emerges from the simultaneous computation of contact forces at multiple points.

For an ideal point contact between tangent surfaces, force computation typically uses a normal in the direction of the penetration depth. When contact surfaces are modeled using multiple contact points between pairs of geometric primitives, such as points, triangles, and edges, the number of point contacts between two objects can be large. This results in costly solves, possibly involving singular systems of equations. Additionally, due to the discretization of time and geometry, the set of contact points can dramatically change from one moment to the next. This is especially true when simulating contact between rigid objects.

We present an alternative for the case of contact between rigid bodies. Our model augments a frictional point contact with rotational torques to permit a resistance to rolling and spinning. Figure 1 shows an example in two dimensions. The square can remain motionless on an immobile supporting surface for a variety of applied body-frame wrenches (combined forces and torques) shown in blue. The square does not move when contact forces can produce the necessary opposing wrench. For the case of rigid bodies, there exists a center of pressure at which the linear contact force, shown in red, is applied to produce the opposing wrench. For a larger body torque, however, there is a point where the linear contact force cannot move past the boundary of the contact area, and the square will start to tip over. With a model of the contact patch, we can include limits on the torques possible between the two contacting bodies.

*e-mail:kry@cs.mcgill.ca

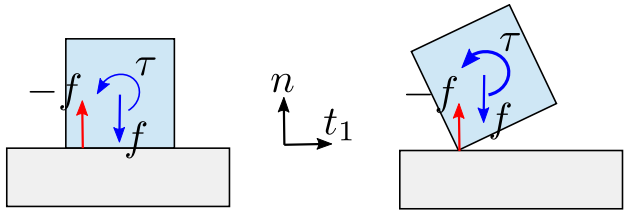

Figure 1: Left: for a body-frame wrench (blue), the center of pressure can shift the linear contact force (red) within the contact surface to produce the opposing torque to prevent rolling. Right: rotation can occur when the center of pressure reaches the boundary of the contact surface.

In the construction of our contact constraints, we follow the classic approach of constraining the motion between two bodies. But instead of constraining the relative velocity at a large collection of contact points, we constrain the 6D relative rigid motion and project the resulting 6D wrench into a region of allowable linear forces and rotational torques (those permissible by the contact patch geometry).

We use a novel technique to model the contact patch between two bodies based on their intersection volume. This builds on other recent work that uses fast image based techniques to compute contact information. Similar to previous work, the first moment of the volume and the volume gradient provide the contact location and normal. However, we also compute the second moment of the volume because this provides an approximation of the shape of the contact patch.

We assume planar contact in this work, and we present an iterative solver that uses the contact patch size and shape to limit torques in a justified manner. We solve block wise on each 6D contact for faster convergence, as opposed to breaking the problem up into multiple contacts or individual constraint components. Finally, our model also includes bounds on torques that oppose spinning, which are enforced in our projection step.

Concretely, our contribution is a new 6D contact model that has the ability to constrain rolling and spinning in conjunction with the traditional Coulomb friction point-contact model. Rolling and spinning are no longer behaviors that emerge only from a collection of contacts. We only need one 6D constraint between each pair of bodies in contact. After providing a review of related work in Section 2, we give a high level overview of our entire method in Section 3. We then explain our new intersection volume computations in Section 4 and provide a complete description of our projection algorithm in Section 5. Finally, we present our results and discuss the advantages and limitation of our method.

2 RELATED WORK

Collision detection and collision response have been extensively studied. A good overview of collision detection is given by Teschner et al. [29]. We briefly review contact force computation methods for rigid bodies, and we provide a more complete, though short, review of previous work on contact modeling.

Baraff [2] proposes a pivoting method to compute contact forces with friction. Stewart and Trinkle [26] turn the problem into an LCP using a linearization of the friction cone. Furthermore, they resolve the Painlevé paradox by using a velocity level formulation. Kaufman et al. present a staggered projection approach [15], which is applicable to both rigid and deformable objects. In slightly earlier work, Kaufman et al. [14] present a fast approximation of frictional dynamics, which models friction in the 6D rigid body configuration space. While our work shares the idea of treating the friction constraint in 6D, our projection step is very different, and our constraints are between pairs of bodies as opposed to treating contact with several bodies simultaneously. The specific case of a single six degree of freedom rigid body colliding with obstacles, without friction, has also been studied in haptics [22].

Since solving a single contact between two rigid bodies is relatively easy [11, 19], iterative methods that process each contact independently of the others are popular [4, 21]. The slow convergence of the sequential approach has been addressed through sweeping [10], parallelization [30], shock propagation [7], and Krylov methods [12, 24]. Our method is similarly based on an iterative solver. We process contacts independently and exploit projection simplifications that follow the work of Hahn [11]. We also take inspiration from work by Tasora and Anitescu on iterative cone complementarity projections [28].

Most of the previous work on contact modeling has focused on reducing the number of contact points for faster force computation. O'Sullivan and Dingliana [23] interrupt collision detection during the traversal of a sphere tree, and use clustered spheres as contact surfaces. In the case of deep penetrations, Erleben [6] prunes all the contacts but the deepest. Additional contact filtering techniques have been proposed for fast interactive simulations and sound synthesis applicatoins [20, 32].

Kaufman et al. [16] avoid multiple redundant contact constraints by producing representative contact samples using a sophisticated combination of geometrical primitives and hierarchies. It is also possible to build a hierarchy of bounding volumes and carefully order the nodes at initialization time so that a good coverage of the contact surface can be obtained using a small subset of points [3,9]. Other reduced or specialized contact models include that of Kry and Pai [17], which deals with spinning, rolling, and sliding for point contact between smooth surfaces through generalized coordinates.

Volume-based contact considers intersection volume rather than penetration depth. The corresponding penalty force can be efficiently computed using layered depth images in three orthogonal directions [8]. Allard et al. [1] build upon this technique with a constraint based formulation for frictional contact, which uses a voxelization of the intersection volume to resolve the constraints at any selected resolution. Wang et al. [31] demonstrate an adaptive image-based technique using ray casting to compute very accurate intersection volumes, allowing for improved simulations of contact. Our computation of intersection volume information is similar, except that we also compute the second moment of the volume as an approximation of the contact patch shape.

Jain and Liu [13] show that physics based character control can be improved through the use of a soft contact model, for instance, during grasping. Having the ability to control the torque at a contact point in a physically justified manner related to the contact patch is important, and similarly a feature of our contact model too. Other recent work has modeled soft contact in haptic simulation with aggregate volume contact constraints [27]. This shares similarities with our method in that these aggregate constraints resist rolling and spinning in a manner consistent with the contact patch geometry. In contrast with our work, their model focuses on contact with deformable fingers, while we work with rigid bodies.

3 6D CONTACT METHOD OVERVIEW

We parameterize the six degrees of freedom of rigid objects with a position and a quaternion, and we describe the velocity of a rigid body using a six component vector containing the linear and angular velocity. As in other multi-body simulators, we refer to the constraints between bodies as joints. That is, we can view unilateral contact constraints as a type of joint, just as we can use bilateral constraints to form mechanisms (for instance, using rotary joints and spherical joints).

Each joint is represented by a set of scalar constraints on the relative motion of the objects. Using six independent constraints produces a rigid joint. At each time step, the object velocities are updated based on the mass and external forces, generating constraint violations. The violations are then iteratively reduced by looping over the joints to compute constraint forces and velocity corrections until the overall residual falls under a given threshold, or a maximum number of iterations is reached. At each joint, a Schur complement equation is solved, specifically,

$$JM^{-1}J^T\lambda = -\mathbf{e} \tag{1}$$

where $\mathbf{e}_{m\times 1}$ is the vector of constraint violations, m is the number of scalar constraints in the joint, the vector of Lagrange multipliers $\lambda_{m\times 1}$ contains the constraint forces necessary to cancel the violations, the Jacobian matrix $J_{m\times 12}$ relates the two object velocities to the constraint violation, and $M^{-1}_{12\times 12}$ is the inverse block-diagonal mass matrix. Given the solution for the Lagrange multipliers, the object velocities are incremented by $J^T\lambda$, which locally cancels the residual. But this also modifies the other joint constraint violations, requiring multiple iterations across the blocks corresponding to each joint. This block Gauss-Seidel method is known to converge in practical cases, but not always very fast when there are multiple redundant contacts between objects. Thus, it is attractive to treat all contact between a pair of objects using a unified 6D constraint in a single block.

Our method projects contact wrenches onto a feasible manifold that depends on the properties of the contact. Given that every interaction between two bodies is represented as a single unified 6D contact, we need to differentiate between different behaviors depending on the situation. That is, we use a rigid joint in Equation 1, but then take care to ensure that the objects in contact can produce sliding, separation, rolling, and spinning. This is done by imposing limits on the different constraint forces, for instance, limiting friction forces depending on the friction coefficient, and limiting torques depending on the shape of the contact patch. We discuss each of these below.

3.1 Feasible frictional forces

Following the joint analogy, a point contact can initially be viewed as a spherical joint where the three translations are only partially constrained so as to allow for separation or frictional sliding. Given an intersection volume between two objects, we initially model the contact as a spherical joint at the middle of the intersection volume. We use the intersection gradient as the contact normal, and with two orthogonal vectors in the tangent plane, we form a coordinate frame $(\mathbf{n},\mathbf{u},\mathbf{w})$ in which we express the residual relative velocity between the two contacting bodies. The force computed by solving Equation 1 cancels all relative velocity and brings this residual to zero.

The force along the normal should only be repulsive, and it should be zero if the objects move away from each other (contact is a unilateral constraint with complementarity conditions, also called Signorini conditions). We use a variant of the cone complementarity approach. If the solution of Equation 1 for the current block does not lie in the friction cone, we restrict the solution space of a new Schur complement equation to the current solution projected

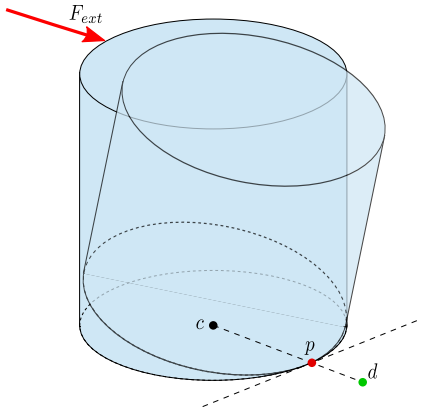

Figure 2: An example of the behavior expected when the center of pressure d is outside the contact patch. The projected center of pressure p is used as the location of a subsequent solve with a point contact to allow for rotational motion.

onto the cone, and solve for a new force. This extra solve, as proposed by Hahn [11], is useful for faster convergence. It can avoid costly iterations by immediately providing a plausible interaction force that respects the non-interpenetration constraint while sliding. However, it does not guarantee that the friction force will perfectly oppose the slip velocity, which may need to be resolved with additional iterations.

3.2 Feasible torques

For a rigid joint, the torque computed in Equation 1 will counteract any rotational relative motion between the two bodies. However, given the contact geometry of certain situations, it is not realistic to remove all rotational motion. We monitor the center of pressure of the contact to evaluate whether rolling should occur or not. Given the wrench between two bodies, we compute the center of pressure as the point on the contact tangent plane where the constraint wrench has zero torque about the tangent directions when expressed in a coordinate frame at this point. Note that spinning torque is treated separately and we will come back to this in a moment.

Consider the case of a cylinder standing on a flat table as shown in Figure 2. Solving Equation 1 for a rigid joint, we get a 6D wrench that will zero the relative motion of the cylinder to the table. We compute the center of pressure of the wrench and check if the point is within the extent of the contact patch. If it is outside of the patch, then we must change the constraint wrench to allow the cylinder to start tipping. To reproduce this behavior when the center of pressure lies outside the patch, we do a new solve with a modified Schur complement using a point contact constraint at the center of pressure location projected to the boundary of the patch. We assume the contact patch shape to be an ellipse. When this is an accurate approximation, we avoid the need of having many contact points around the boundary, as we can instead use a projection to approximate the point about which the object will tip. However, there are some important subtleties to this projection. We present the details in Section 5.

Constraints on spinning torques are simpler. When the center of pressure is within the contact patch, spinning torques are clamped based on the normal force and the distance to the boundary. When the center of pressure approaches the edge of the boundary, we recognize that the pressure distribution across the patch goes to zero. Thus, points away from a center of pressure at the boundary will not be able to provide frictional forces to resist spinning.

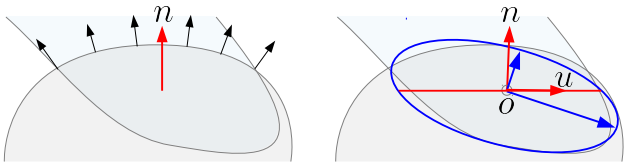

Figure 3: Side view of the geometrical model of the contact for an exaggerated large interpenetration volume. Left: intersection of two objects, with arrows representing the intersection volume gradient on one object, and the normal direction set to be parallel to the net corresponding pressure force. Right: contact volume modeled using an ellipsoid with principal axes shown in blue. The ellipse in the contact plane is shown in red, and the contact coordinate frame is placed at the point in the center.

4 Contact Geometry

After a discrete-time integration step, colliding objects generally intersect each other and the idealized point contact model typically used in contact force computation is not directly suitable. Instead of approximating a contact surface using multiple contact points, we exploit the shape of the intersection volume to capture the behavior of the interaction using a unified contact model with rotational constraints. We take inspiration from biped simulation, where it is well known that static equilibrium is possible only if the center of mass of the biped is vertically projected inside the convex hull of the foot-ground contact points. The information provided by this convex hull is easier to exploit than a set of independent contacts to infer the stability. Computing convex envelopes is easy for a small number of points, but complex for a potentially large number of intersection points in space.

4.1 Intersection volume and gradient

The mono-volume contact model discussed by Allard et al. [1] creates one contact model per intersection volume and efficiently computes the average normal direction. Parallel rays are cast on the GPU through the object surfaces, and for each ray, a simple one-dimensional intersection test is used to detect crossings and to compute the approximate intersection volume. Doing this in three orthogonal directions allows the computation of the partial derivatives of the intersection volume with respect to the vertex coordinates. These correspond to pressure forces that are parallel with the vertex normals, and their net sum on each object defines the direction of the overall contact normal, as illustrated in the left of Figure 3.

4.2 Intersection shape

We extend the image-based mono-volume model to collect information about the shape of the intersection volume. The difficult part is the efficient modeling of the shape based on ray-surface intersections. The points identified in this process are too numerous to efficiently compute a convex hull. We thus use a simplifying assumption: we model the contact surface as a ellipse, and we compute the major axis of this ellipse using the second moment of the intersection volume. This tensor is much like the inertia tensor, but in this case it does not account for varying density. Since the rays are parallel, the intersection volumes associated with each ray are a partition of the total volume, and the net tensor is the sum of each ray's contribution.

This 3×3 symmetric positive definite tensor provides us with a basic model of the shape and orientation of the contact surface. Its eigenvalues are related to the dimensions of the ellipsoidal approximation of the intersection volume, and its eigenvectors provide us with its orientation, as illustrated in blue in the right of Figure 3. As we approach a zero volume intersection with a perfectly tangent

contact between flat surfaces, one eigenvector will be parallel to the normal and the corresponding eigenvalue will be equal to the sum of the other two (see Appendix A). The two other directions provide us with an ellipse-shaped model of the contact area. In the general case, after a time integration step, the contact volume is neither zero nor flat and the ellipsoid may not be perfectly aligned with the normal. In this case, we project the ellipsoid to a flat ellipse in the tangent plane defined by the contact normal. Note that it is always the linear component of the intersection volume gradient that we use as the normal.

5 Contact Force and Torque

We use a rigid joint with six scalar constraints to compute the first guess of the contact wrench. This is equivalent to assuming a sticking contact with no relative rolling or spinning motion. We solve Equation 1 with six scalar constraints expressed in the coordinate frame at the center of the intersection volume (as described in Section 3.1). The result is a wrench that contains both a 3D force and a 3D torque. Given this solution, we may need to further modify the force and torque to satisfy our contact constraints.

5.1 Schur complement and adjoint matrices

Equation 1 represents the constraint of one contact at the center of the patch. During the projection process, we need to move the constraints into another reference frame by reformulating the constraint equation. To do so, we compute the adjoint matrix, which is used to transform the wrenches and velocities from one reference frame to another. For example, the adjoint that transforms velocity from a world frame to a contact frame is written

$$ {}^{c}_{w}\mathsf{Ad} = \begin{bmatrix} {}^{c}_{w}R & \hat{\mathbf{p}}\,{}^{c}_{w}R \\ 0 & {}^{c}_{w}R \end{bmatrix} \tag{2} $$

where ${}^{c}_{w}R$ is the rotation matrix, $\mathbf{p}$ is the translation, and $\hat{}$ denotes the cross product operator. To transform velocities from one frame to another we multiply by the adjoint, whereas to transform wrenches, we use the inverse transpose of ${}^{c}_{w}\mathsf{Ad}$. Note that when using the adjoint defined above, the wrench and velocity vectors have their linear quantities in the top three components, and rotational quantities in the bottom three.

Given the Schur complement block for a contact expressed in coordinates of the world frame, we can easily produce an equivalent constraint problem in different coordinates by multiplying on both sides. That is,

$$ {}^{c}_{w}\mathsf{Ad}\; JM^{-1}J^{T}\,{}^{c}_{w}\mathsf{Ad}^{T}\lambda_{c} = -\,{}^{c}_{w}\mathsf{Ad}\;\mathbf{e} \tag{3} $$

is equivalent to Equation 1, except that the solution is now expressed in a contact reference frame. Writing the problem or the solution in different frames is useful as we can easily monitor important quantities, such as tangential sliding, or make simple modifications to the system, for instance, to remove a rolling constraint.

5.2 Cone projection

The contact force is decomposed into a repulsion force $\mathbf{f}_n$ parallel to the normal and a tangential force $\mathbf{f}_t$ in the tangent plane. It is projected to the Coulomb cone centered on the normal, as presented in Section 3.1. If the force is outside the Coulomb cone, we rewrite Equation 1 by constraining the linear component of the spatial force to be in the direction of the Coulomb cone projection while keeping the non-rolling assumptions of a rigid contact. We can write this as

$$ RJM^{-1}J^{T}Q\lambda' = -R\mathbf{e}, \tag{4} $$

where

$$ Q = \begin{bmatrix} \mathbf{q} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \in \mathbb{R}^{6\times 4}, \tag{5} $$

$$ R = \begin{bmatrix} \mathbf{e}_1^{T} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \in \mathbb{R}^{4\times 6},\ \mathbf{e}_1^{T} = [1\,0\,0]. \tag{6} $$

Here, $\mathbf{I}$ is a 3×3 identity matrix, and the column vector $\mathbf{q}$ in Equation 5 is the direction of the force projected onto the cone. After solving Equation 4, the wrench is recomputed as $Q\lambda'$. Solving for new values of torques is essential, as there is coupling between the force and torque parts of the wrench.

An intuitive example that can be used to illustrate this is a sphere landing on an incline plane with zero friction. The sphere is expected to land and slide without rolling. The initial force computed is directly opposing the gravity, and the computed torque opposes the moment produced by this force. Given that there is no friction, the cone of permissible forces is a ray in the normal direction, and the linear force obtained by resolving the system will be along the normal. But the wrench we initially computed included a torque that opposed the moment produced by the non-normal force (that is, the force was not pointing toward the center of mass of the sphere). If we were to keep the torque and only project the force, we would get a net moment around the center of the sphere, and the sphere will begin to roll. Since the center of the contact patch and the center of mass of the sphere form a line parallel to the normal, the new solution will not produce any moment and the computed torque will be zero. This observation is particularly important as we use the torque to compute the location of the center of pressure as explained below.

5.3 Rolling torque and center of pressure

Identifying the center of pressure of a contact wrench is important to determine if two objects will exhibit rolling behavior. The adjoint inverse transpose can let us express wrenches in different coordinate frames, but we can also build intuition by looking at a simpler equation. Given a force $\mathbf{f}$ and a torque τ_c expressed at a point $\mathbf{c}$, the equivalent wrench expressed at point $\mathbf{p}$ has the same linear force, and has the torque

$$ \tau_p = \tau_c + (\mathbf{p}-\mathbf{c})\times\mathbf{f}. \tag{7} $$

We decompose the contact torque τ into a normal component τ_n opposed to spinning, and a tangential component τ_t opposed to rolling. The center of pressure (COP) is the point in the contact plane where the equivalent wrench has zero tangential torque. We can think of this as the location of the repulsion force. Given the rolling torque $\tau_t = (\tau_u, \tau_w)^{T}$ in the contact frame, the COP is located at $\mathbf{d} = \frac{1}{\mathbf{f}_n}(\tau_w, -\tau_u)^{T}$, as illustrated in Figure 4.

5.4 Center of pressure projection

If the COP lies inside the contact area modeled by the elliptical contact patch, the rolling torque is feasible and we leave it unmodified. Otherwise, it corresponds to a pressure distribution which can not be positive everywhere in the contact area, which violates the Signorini conditions. We thus clamp it to the closest edge, for example, point $\mathbf{p}$ in Figure 4, and the resulting rolling torque becomes $\mathbf{f}_n \times (\mathbf{p}-\mathbf{c})$. This can be done with an iterative root finding algorithm. A good solution can be found in 4 to 5 iterations.

We use a Euclidean distance metric in the projection, and observe good behavior in practice. Nevertheless, for frictionless contact, we note that the projection should instead use a mass weighted metric that takes into account the inertia of the two objects in contact. However, for frictional contact, there does not exist a simple metric that will allow us to use projection to predict the center of pressure location. Instead, we choose the simple Euclidean projection and allow errors to be resolved through multiple iterations in

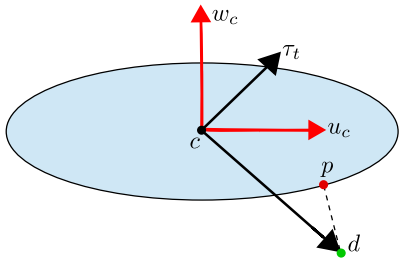

Figure 4: Computing and projecting the center of pressure. Based on the contact force $\mathbf{f}$ and torque τ at the volume center $\mathbf{c}$, we compute the center of pressure. If it lies outside the approximate contact patch ellipse, we project it to the closest point.

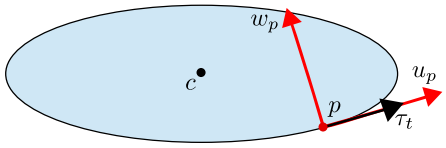

Figure 5: After the projection step, the contact frame is oriented to align with the edge of the patch.

the solver, or ultimately with subsequent time steps and changing intersection volumes.

5.5 New coordinate frame

We assume that rolling will occur when the center of pressure is projected onto the edge of the patch. In this case, we transform the fixed joint into a point-contact joint located at the projected position. We do this by rewriting Equation 1 as a 3-by-3 subsystem where there is no constraints on the angular velocity,

$$PJM^{-1}J^TP^T\lambda = -P\mathbf{e}, \tag{8}$$

where

$$P = \begin{bmatrix}\mathbf{I} & 0\end{bmatrix} \in \mathbb{R}^{3\times 6}. \tag{9}$$

The solution of this new Schur complement equation gives us a new wrench, which consists only of a linear force when viewed in the new projected center of pressure reference frame. This frame is aligned with the tangent of the projection ellipse as shown in Figure 5. This realignment allows us to check if the wrench and velocity corresponds to simple rotation about the tangent or if it otherwise corresponds to more complex behavior. Note that in the new solution the linear force is changed, allowing the whole system to respect the non-separation assumption. Moving the force back to the center of the patch will produce non-zero torque, but this will be feasible given the geometry of the patch.

At this point, we must verify that the newly computed linear force still lies inside the friction cone. If it does not, we repeat the cone projection described in Section 5.2, with the difference being that we have a spherical joint with no constraints on the torque (that is, we combine Equations 4 and 8). The system effectively becomes a 1-dimensional equation, where the only Lagrange multiplier to be computed is the magnitude of the repulsion force along the side of the cone. The system we solve is

$$R'PJM^{-1}J^TP^TQ'\lambda = -RP\mathbf{e}, \tag{10}$$

where $R' = \mathbf{e}_1 \in \mathbb{R}^{1\times 3}$ and $Q' = \mathbf{q} \in \mathbb{R}^{3\times 1}$, as defined in Section 5.2.

5.6 Spinning torque

The spinning torque is generated by tangential forces distributed on the contact patch, which are bound by Coulomb friction. Computing the maximum spinning torque requires knowledge of the normal force distribution across the contact area, which is not available. Computing this distribution is an ill-posed problem which requires simplifying assumptions. We propose to assume that the contact force is distributed on the four corners $(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4)$ of the rectangle enclosing the elliptical patch according to the generalized barycentric coordinates of $\mathbf{p}$ in the rectangle,

$$\mathbf{p} = \sum_{i=1}^{4} w_i \mathbf{c}_i, \tag{11}$$

$$\mathbf{f}_i = w_i \mathbf{f}. \tag{12}$$

We use the product of the bilinear coordinates as weights. We assume that there is no spinning torque if the center of pressure was outside the patch because this situation is equivalent to a pressure distribution concentrated on a single point. This force distribution is physically consistent, since the virtual work of the force $\mathbf{f}$ applied to $\mathbf{p}$ is equal to the virtual work of the distributed forces: $\mathbf{f}\cdot d\mathbf{p} = \sum_i \mathbf{f}_i \cdot d\mathbf{c}_i$.

Accurately computing the maximum spinning torque created by these four points would require an optimization with linear constraints to leave the net tangential force unchanged, and non-linear unilateral constraints for Coulomb friction. Instead of solving this complex problem, we compute the upper bound of the torque using the maximum tangential force on each point times the distance to the center of pressure,

$$\tau_p \leq \mu \sum_i f_{ni} \|\mathbf{c}_i - \mathbf{p}\| \tag{13}$$

where f_{ni} is the normal force at contact $\mathbf{c}_i$ and μ is the Coulomb friction coefficient. This approximate model provides qualitatively reasonable behavior, though it can over estimate the torque available to resist spinning.

An overview of all steps of our algorithm as described in Sections 5.1 to 5.6 is summarized in Algorithm 1.

5.7 Stabilization

Our method relies on the existence of some interpenetration between the objects. Problems can occur when the intersection volume is a thin sheet or a skinny rod. In the first case, a small motion could lead to big variations in our selection of the center of the patch. In the second case, computing the normal is difficult as the volume gradient is constant for the plane tangential to the long axis of the rod.

To reduce the occurrence of these limit cases, we introduce a combination of compliance and stabilization into the solver, which allows the contacts to conserve enough interpenetration to get a good volume measurement, and therefore a good patch approximation. Post-stabilization [5] is done at the position level in the direction of the volume gradient. We use a parameter e as a measurement of the target interpenetration volume and we tune this parameter so that it is appropriate for the scene. This means that the stabilization will move the objects in the direction in which the volume reduction is the greatest.

6 Results and Discussion

We implemented our method in C++ in the SOFA Framework [25]. We reused the plug-in that Wang et al. [31] developed for adaptive intersection volume computation and added the inertia tensor computations. This ray casting plug-in uses the NVidia Optix library written in CUDA. The contribution to the inertia from each ray is summed on the GPU. Using SOFA allows us to compare

Algorithm 1 Iterative contact force computation.

Input: Fixed joint Lagrange multipliers
Output: Feasible Lagrange multipliers

function PROJECT(λ_w)
  $\lambda_c \leftarrow {}^{c}_{w}\mathrm{Ad}^{-T}\lambda_w$ *Solution in contact frame*
  $S_c \leftarrow {}^{c}_{w}\mathrm{Ad}\ JM^{-1}J^T\ {}^{c}_{w}\mathrm{Ad}^T$ *System in contact frame*
  $\mathbf{e}_c \leftarrow {}^{c}_{w}\mathrm{Ad}\ \mathbf{e}$ *(see §5.1)*
  if $\mu\lambda_n \leq |\ \lambda_t\ |$ **then**
    $\mathbf{q} \leftarrow$ CONEPROJECTION(λ_c, μ) *Friction direction*
    $\lambda \leftarrow$ SOLVE($RS_cQ, -R\mathbf{e}_c$) *(see §5.2, Eq. 4)*
  end if
  $\mathbf{d} \leftarrow \frac{1}{\mathbf{f}_n}(\tau_w, -\tau_u)^T$ *Compute COP (§5.3)*
  $\mathbf{p} \leftarrow$ COPPROJECTION($\mathbf{d}$) *Patch boundary proj. (§5.4)*
  if $\mathbf{d} != \mathbf{p}$ **then**
    $S_p \leftarrow {}^{p}_{c}\mathrm{Ad}\ S_c\ {}^{p}_{c}\mathrm{Ad}^T$ *System in projected frame*
    $\mathbf{e}_p = {}^{p}_{c}\mathrm{Ad}\ \mathbf{e}_c$
    $P \leftarrow \left[I^{3x3}; 0\right] \in \mathbb{R}^{6x3}$ *Point contact conversion*
    $\lambda_p \leftarrow$ SOLVE($PS_pP^T, -P\mathbf{e}_p$) *(see §5.5, Eq. 8)*
    if $\mu\lambda_n \leq |\ \lambda_t\ |$ **then**
      $\mathbf{q} \leftarrow$ CONEPROJECTION(λ_p, μ) *New friction dir.*
      $\lambda_p \leftarrow$ SOLVE($RPS_pP^TQ, -RP\mathbf{e}_p$) *(§5.5, Eq. 10)*
    end if
  else
    $\lambda_p \leftarrow \lambda_c$ *Otherwise no modification*
  end if
  $\tau_n \leftarrow$ SPINCLAMP($\mathbf{p}, f_n, \mu$) *Spinning friction (see §5.6)*
  $\lambda_w \leftarrow {}^{c}_{w}\mathrm{Ad}^T\ {}^{p}_{c}\mathrm{Ad}^T\lambda_p$ *Conversion to world frame*
  return λ_w
end function

our method against other time-stepping algorithms and lets us use different solvers with the 6D contact. All the presented results were simulated using projected block Gauss-Seidel. However, note that our approach is compatible with other iterative LCP solvers based on a force projection, for instance, the solver proposed by Silcowitz-Hansen et al. [24].

6.1 Simulation results

Our method produces realistic motion for a variety of different scenarios. This includes cubes balancing on edges, spheres rolling down incline planes, cube stacking, grasping, and a tumbling cylinder on a plane. We vary initial conditions and friction parameters to explore different behaviors. We also adjust the allowable interpenetration volume, the compliance, and damping based on the contents of the scene. In this section, we present these examples and discuss the particularities of each scene.

6.1.1 Cubes on an edge

Figure 6 shows multiple cubes landing on the edge of a plane with different center of mass positions. Although all the contacts are represented with a single 6D block, different behaviors emerge depending on if the center of mass is above the plane, or how far it is over the edge. Note that the closest cube clears the edge of the plane and slides perfectly down the side of the vertical wall.

6.1.2 Stacking

Simple vertical stacking is a good test to validate contact simulations, where redundant constraints can produce numerically challenging LCPs for many solvers. In our case, we avoid the redundancy and have a simple chain of constraints which is much easier to solve. However, the behavior is dependent on the center of pressure location and a good normal approximation (from the intersection volume gradient). A stable stack needs accurate normals

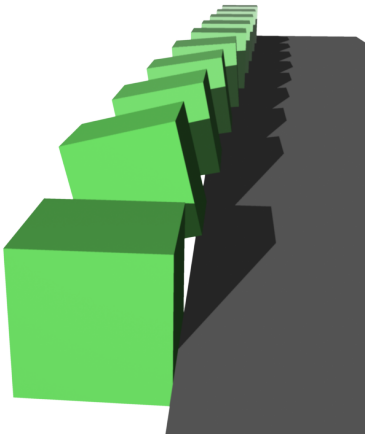

Figure 6: Multiple cubes with different initial conditions landing on the edge of a plane. Those with a center of mass above the plane will stay balanced whereas the other tip over and fall. The closest cube falls vertically because it is far enough from the edge to clear the plane and slides perfectly against the vertical wall.

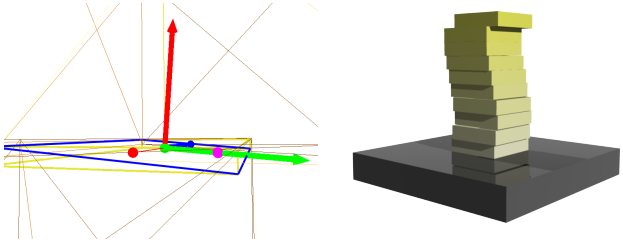

Figure 7: Left: A visualization of the contact volume of a cube falling on the edge of another. The yellow box is the contact volume and the blue rectangle represents the contact plane, which is tilted. Right: A resting stack of cubes with some overhanging parts. In this scenario, the normals are not perfectly aligned with the sides of the cubes.

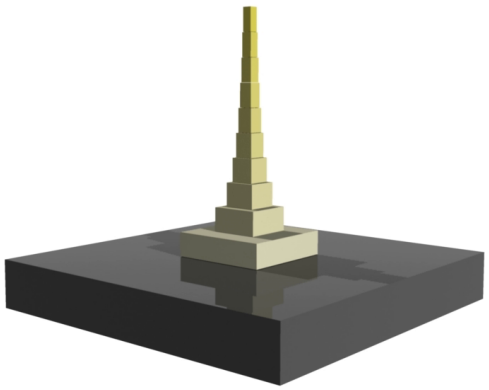

Figure 8: A resting nested stack of cubes is a test case we use to demonstrate the stability of solves using our contact model.

and we noticed that this can be harder to achieve if the intersection volume includes object boundaries, for instance if we have a sloppy stack of cubes, as seen in Figure 7. Specifically, there can be a small lateral component to the intersection volume gradient recognizing that the volume can decrease not only with normal displacement, but also with tangential motion. We note that the smallest inertia

direction corresponds to what we would consider the true normal to the surface, but the intersection gradient is typically close enough so as not to cause problems provided there is some friction in the system. Thus, the stack in Figure 8, specifically avoids this issue through the use of a tapered stack shape. In contrast, we explored low friction settings with a cube dropped from a small height near the edge of another. While the intersection volume and gradient (i.e., the contact normal) depend on the relative velocity and time step, for this test we observe sliding artifacts only when the friction coefficient is below 0.031 (see left of Figure 7 and the supplementary video).

In the future, we could use a heuristic to choose between the volume gradient and the smallest inertia direction for our normal, or use a blended combination of the two.

6.1.3 Rolling sphere

An interesting example where the strong relationship between torques and forces influences the projection is a rolling sphere. In this example, the contact patch is small and the center of pressure projection allows for only a small torque in comparison to the torque required to cancel the velocity, and therefore the sphere will roll. Figure 9 shows the patch and the center of pressure projection. An interesting behavior is the case where the friction is zero. This modification to the initial condition allows us to validate that the computed torque is effectively zero when projecting the linear force along the normal, making the sphere slide without rolling. The two different behaviors can be seen in the supplemental video. Another interesting aspect is the resistance to rolling our method induces naturally depending on the size of the patch. In a perfect scenario, the contact patch would be a point and the torque would be projected to zero. In reality, the small patch still allows some torque to resist the rolling motion and this slows it down a little. The more interpenetration we allow, the more resistance it will create, allowing a behavior similar to a deflated toy ball rolling down a plane.

6.1.4 Spinning sphere

The spinning sphere example shown in Figure 10 demonstrates how a spinning contact comes to rest due to the dry friction spinning torque constraint. We give the sphere an initial rotational velocity along the vertical direction and place it on a planar surface. Depending on the amount of allowed interpenetration, the size of the patch will vary, which leads to a different limits on the spinning torque. Analogous to the case of the rolling sphere, a deflated ball will come to rest faster than a perfect rigid sphere.

Our method uses many assumptions to project the spinning torque. There exist more complex models that could be adapted to our method. Analogous to the friction cone method, the Coulomb Contensou friction model defines a permissible set of tangent friction forces and spinning torques. Leine and Glocker [18] describe this as a friction *ball*. A typical scenario where this effect is prominent is a rotary floor cleaner. While the machine is hard to move when the cleaning pad is not rotating, the tangential friction is greatly reduced as soon as there is a fast rotational motion, but this comes in conjunction with a strong spinning friction torque.

6.1.5 Tumbling cylinder and rolling coin

While a cylinder at rest that is pushed hard enough to tip is an interesting contact transition, the case of a spinning coin coming to rest is another. In the latter, as shown in Figure 11, a small contact patch moves quickly around the outer edge of the coin as it spins. Eventually, it comes to the point of forming a larger contact patch that can provide a wrench that cancels the velocity. The fact that our center of pressure is projected to the closest edge, as opposed to being projected toward the center of the patch, allows such motion to take place. With the projected point acting like a spherical

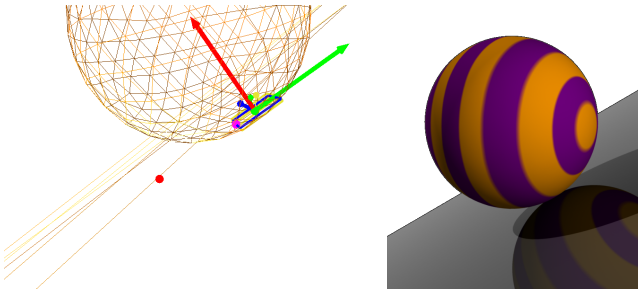

Figure 9: A sphere rolling down an incline plane. The center of pressure in red projects to the pink dot at the boundary of the patch. Note that the intersection inertia is visualized as a yellow wire box.

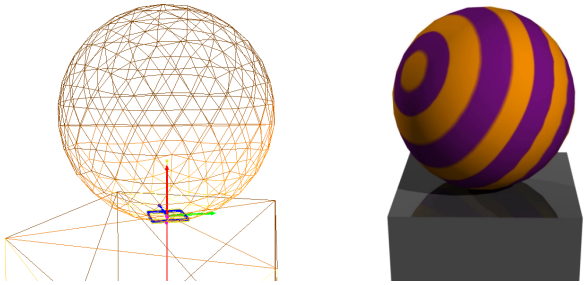

Figure 10: A sphere spinning on a cube. The red vector is the constraint torque, which acts in the normal direction.

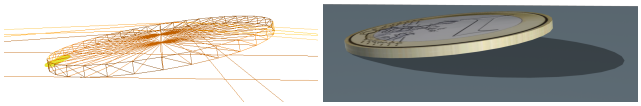

Figure 11: A rolling coin falling down on its edge.

joint, the coin can continue a complex spinning and rolling motion until the patch grows large enough to force the coin to come to rest.

6.1.6 Grasping

The grasping example shown in Figure 12 demonstrates how the compliance and the torque constraints can act together to realistically maintain a firm grip on a ruler. The fingers are attached to rotary joints and their motion and grasping force is produced with a simple proportional derivative control. The stabilization parameters used in this example have an important effect on the observed behavior because the curved finger surfaces produce larger contact patches as interpenetration increases, which directly influences the amount of spinning torque available for holding the ruler horizontally.

6.1.7 Complex geometry

We have compared our method with a point-contact method in the case of a geometrically complex models as seen in Figures 13 and 14. The scene in Figure 13 consists of a bumpy cube landing on a plane with both normal and tangential velocities. In this scenario, the roughness of the cube causes a large number of point contacts to be created. While the example may be somewhat contrived, it

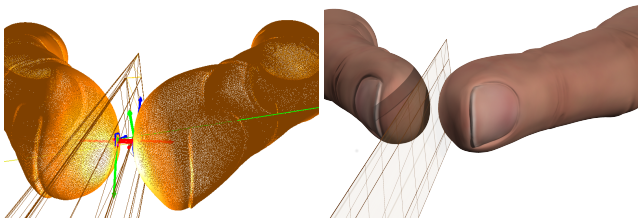

Figure 12: Pinch grasp of a transparent ruler with 6D contact. The combination of compliance and torque constraints gives a realistic pinch grasp that emulates the behavior of real human fingers.

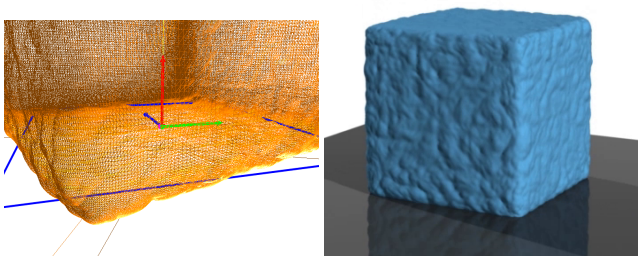

Figure 13: A bumpy cube on a plane only needs one 6D contact, in comparison to a large number of point contacts.

Figure 14: Armadillo salad simulation demonstrating 6D contacts with complex geometry.

does help us make the point that the complexity of 6D contact remains fixed. For simple proxies, our intersection volume processing will take longer than various fast and popular collision detection methods. But when geometries are complex, it can be simpler and faster to use GPU intersection volume computations, and the single 6D contact constraint allows us to avoid large numbers of contact points and redundant constraints. Table 1 shows a comparison of the computation time for this complex geometry scenario along with other scenes. Of particular note is the armadillo salad seen in Figure 14, as this scene involves numerous contacts with different normals between complex geometries.

In our implementation, collision detection takes up the bulk of the computation time, notably due to how we collect the second order moments. We note that code optimization is possible, but we could also use alternate methods to compute the intersection moments, or similarly rely on different techniques for approximating the contact patches.

Scene (Triangles)	Method	Detection	Assembly	Factorization	Solve
Cube (197k)	6D	129.4	26.8	0.05	1.0
	Point-based	86.3	93.3	4402.7	1194.4
Salad (102k)	6D	270.6	0.07	0.11	3.1
	Point-based	481.1	10.6	28.8	50.35
Sphere (524)	6D	180.2	0.02	0.04	0.12
	Point-based	2.31	0.15	0.17	0.31
Stack (264)	6D	228.2	0.16	0.29	186.0
	Point-based	39.9	2.04	7.36	407.1

Table 1: Comparison between our method and a point-based method using the same dynamic solver. The time in milliseconds spent on different sections of the algorithm per timestep is shown for the bumpy cube example, the armadillo salad, the rolling sphere and the stable stack. Timings were done on a machine with a i7 8-core CPU, 8GB of RAM, and a GForce GTX Titan GPU.

	6D Method		Point-based method	
Scene	Iteration count	Residual	Iteration count	Residual
Stack (197k)	74±0	2.8±0.3 e-16	3087±260	3.5±0.8 e-15
Salad	40±23	1.6±1 e-16	1796±3512	2.7±8.2 e-15

Table 2: Comparison between our method and a point-based method using a projected Gauss-Seidel solver. For both methods we recorded the number of iterations and the residual for 100 time steps and compute means and standard deviation. Note that the point-based method on the salad scene shows a high variation in the error and the number of iterations.

6.1.8 Convergence

Point-based methods can use contact pruning to reduce the number of redundant contacts between the objects. One of the advantage of our method is that redundant contacts need not be created in the first place. Iterative methods such as projected Gauss-Seidel can converge much slower when a system is over constrained. Our method reduces this poor convergence behavior because only one contact is created per pair of objects. Also, contact persistence is high, making warm starts easy to use and further improving the convergence rate.

We compared the convergence of our method versus a vanilla point-based method for the stack and the salad scene. Table 2 shows the iteration counts for both methods given a target stopping criterion. The criterion we use is the 2-norm of the update vector constructed during a single Gauss-Seidel iteration. As such, the first entries of the update vector are based on the constraints of the previous iteration step, whereas the last update entries are based on the updated values. This has the advantage of not requiring a matrix-vector multiplication to compute the stopping criterion alone, taking advantage of the block structure.

In Figure 15, we show the value of the stopping criterion at each iteration for both scenes. The solves are started without warm start at a time step where the geometries are in resting contact. It is important to note the large difference in the number of constraints between the scenes. This does influence our comparison in Table 2 and the convergence plots because the number of terms in norm of the update vector, and how each component measures error (either penetration depth or penetration volume). Furthermore, we note that the number of terms (i.e., constraints) has a big effect on computational cost of each iteration.

A typical problem that causes convergence issues in LCP solvers is when the presence of redundant constraints and friction. In those conditions, there exist many valid solutions that distribute the forces differently across the constraints, which can cause cycling issues and impact the speed of convergence. To compare our method with the point-based method in such a scenario, we

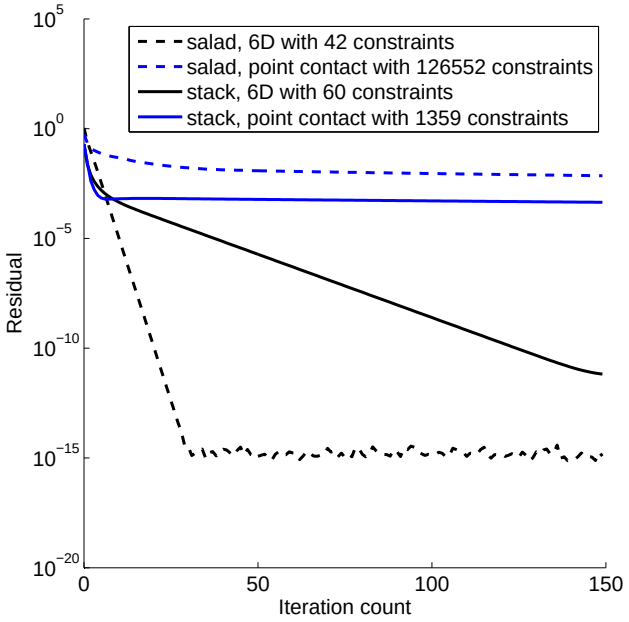

Figure 15: The 2-norm of the accumulated constraint update vector at each iteration. The constraint update in blue is from the point based method and the lines in black represent our method.

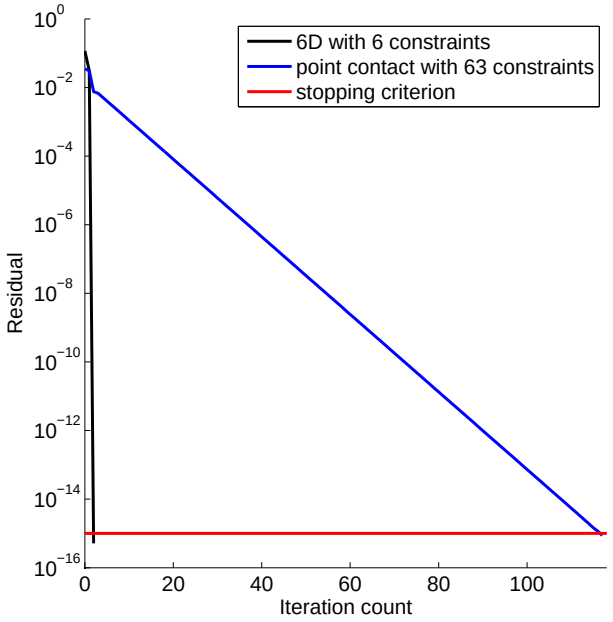

Figure 16: The 2-norm of the accumulated constraint update vector at each iteration for an octagonal sliding block. The 6D method is shown in black and the point-based method in blue.

simulated a simple octagonal block sliding on a plane with dry friction. The stopping criterion was set at 1×10^{-15} for both the point-based method and the 6D method. The point-based method generates 21 contact points, or 63 constraints, whereas the 6D method only needs 6 scalar constraints. Figure 16 shows that the 6D method reaches the threshold in only 3 iterations, while the point-based method takes 118 iterations.

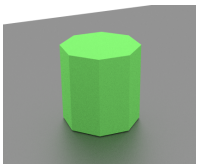

Figure 17: Octagon sliding on plane with friction.

6.2 Limitations and future work

Using an ellipse as a contact patch model is a simple approximation of the contact as identified by the intersection volume. It will underestimate the size of the patch in the case of non-convex contact surfaces. For instance, the four legs of a table standing on the ground produce a smaller patch size than we would have with the convex hull of the intersection volume. While the convex hull is difficult to compute for complex geometries, it may be an attractive alternative to explore in future research. Along these lines, we have explored the use of k-DOP approximations of the intersection volume, as they can be computed quickly using our ray casting implementation.

In practice, we note that when poor contact patch approximations occur during a simulation, it is typically only for a brief time. Stepping the simulation will often move the objects in a way that produces different interpenetration volumes resulting in better contact patch approximations.

Another important limitation with our approach is that we currently only treat planar contact. It is arguable that most scenarios involve non-planar contact. Nevertheless, we note that our unified 6D contact model can still produce plausible interaction forces in many cases, and the armadillo salad simulation is a good example of this. Segmenting the intersection volume into planar components prior to computing our patch approximation would allow for higher accuracy.

7 Conclusions

Our method leverages the information produced by a volume-based collision detection method to constrain rolling and spinning torque in addition to a point-contact Coulomb friction model. It allows realistic rolling and spinning behaviors to arise from a single unified 6D contact representation between object pairs. It has the advantage of eliminating redundant contacts and reducing the size of the system to solve. Our 6D contact model is not restricted to sequential force computation methods, and using it in parallel solvers or with Krylov methods is an obvious avenue of future work.

Acknowledgements

The authors wish to thank Laura Paiardini for a help with the renderings and the video editing. We also thank the anonymous reviewers for the suggestions for improving the paper. This work was supported by funding from NSERC, CFI, GRAND NCE, and Fonds unique interministériel Dynam'it!.

References

[1] J. Allard, F. Faure, H. Courtecuisse, F. Falipou, C. Duriez, and P. G. Kry. Volume contact constraints at arbitrary resolution. *ACM Trans. Graph.*, 29:82:1–82:10, July 2010.

[2] D. Baraff. Fast contact force computation for nonpenetrating rigid bodies. In *Proceedings of SIGGRAPH '94*, Computer Graphics Proceedings, Annual Conference Series, pages 23–34. ACM SIGGRAPH, ACM Press, July 1994.

[3] J. Barbič and D. L. James. Six-dof haptic rendering of contact between geometrically complex reduced deformable models. *EEE Trans. Haptics*, 1(1):39–52, Jan. 2008.

[4] Bullet Physics. www.bulletphysics.org.

[5] M. B. Cline and D. K. Pai. Post-stabilization for rigid body simulation with contact and constraints. In *IEEE ICRA*, 2003.

[6] K. Erleben. Contact graphics in multibody dynamics simulation. Technical report, Technical University of Copenhagen, 2004.

[7] K. Erleben. Velocity-based shock propagation for multibody dynamics animation. *ACM Trans. Graph.*, 26(2):12:1–12:20, June 2007.

[8] F. Faure, S. Barbier, J. Allard, and F. Falipou. Image-based collision detection and response between arbitrary volumetric objects. In *Pro-*

ceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '08, pages 155–162, 2008.

[9] L. Glondu, S. C. S. M. Marchal, G. Dumont, and M. A. Otaduy. Efficient collision detection for brittle fracture. In *ACM/Eurographics Symposium on Computer Animation*, 2012.

[10] E. Guendelman, R. Bridson, and R. Fedkiw. Nonconvex rigid bodies with stacking. *ACM Trans. Graph.*, 22(3):871–878, July 2003.

[11] J. K. Hahn. Realistic animation of rigid bodies. In J. Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 299–308, Aug. 1988.

[12] T. Heyn, M. Anitescu, A. Tasora, and D. Negrut. Using krylov subspace and spectral methods for solving complementarity problems in many-body contact dynamics simulation. *IJNME*, to appear, Preprint ANL/MCS-P2099-0612, 2013.

[13] S. Jain and C. K. Liu. Controlling physics-based characters using soft contacts. *ACM Trans. Graph.*, 30(6):163:1–163:10, Dec. 2011.

[14] D. M. Kaufman, T. Edmunds, and D. K. Pai. Fast frictional dynamics for rigid bodies. *ACM Trans. Graph.*, 24(3):946–956, July 2005.

[15] D. M. Kaufman, S. Sueda, D. L. James, and D. K. Pai. Staggered projections for frictional contact in multibody systems. *ACM Trans. Graph.*, 27(5):164:1–164:11, Dec. 2008.

[16] D. M. Kaufman, S. Sueda, and D. K. Pai. Contact trees: adaptive contact sampling for robust dynamics. In *ACM SIGGRAPH 2007 sketches*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[17] P. G. Kry and D. K. Pai. Continuous contact simulation for smooth surfaces. *ACM Trans. Graph.*, 22(1):106–129, Jan. 2003.

[18] R. Leine and C. Glocker. A set-valued force law for spatial coulomb-contensou friction. *European Journal of Mechanics - A/Solids*, 22(2):193 – 216, 2003.

[19] B. Mirtich and J. Canny. Impulse-based simulation of rigid bodies. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, I3D '95, pages 181–ff., New York, NY, USA, 1995. ACM.

[20] A. Moravanszky and P. Terdiman. Fast contact reduction for dynamics simulation. *Game Programming Gems*, 4:253–263, 2004.

[21] NVidia PhysX. www.geforce.com/hardware/technology/physx.

[22] M. Ortega, S. Redon, and S. Coquillart. A six degree-of-freedom god-object method for haptic display of rigid bodies with surface properties. *IEEE Trans. Vis. Comput. Graph.*, 13(3):458–469, 2007.

[23] C. O'Sullivan and J. Dingliana. Real-time collision detection and response using sphere-trees. In *Spring Conference on Computer Graphics (SCCG'99)*, 1999.

[24] M. Silcowitz-Hansen, S. Niebe, and K. Erleben. A nonsmooth nonlinear conjugate gradient method for interactive contact force problems. *Vis. Comput.*, 26(6-8):893–901, June 2010.

[25] SOFA. http://www.sofa-framework.org.

[26] D. Stewart and J. Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal of Numerical Methods in Engineering*, 39:2673–2691, 1996.

[27] A. Talvas. *Bimanual Haptic Interaction with Virtual Environments*. PhD thesis, INSA Rennes, 2014.

[28] A. Tasora and M. Anitescu. A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics. *Comput. Methods Appl. Mech. Eng.*, 200(5-8):439–453, 2011.

[29] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrman, M.-P. Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser, and P. Volino. Collision Detection for Deformable Objects. *Computer Graphics Forum*, 24(1):61–81, 2005.

[30] R. Tonge, F. Benevolenski, and A. Voroshilov. Mass splitting for jitter-free parallel rigid body simulation. *ACM Trans. Graph.*, 31(4):105:1–105:8, July 2012.

[31] B. Wang, F. Faure, and D. K. Pai. Adaptive image-based intersection volume. *ACM Trans. Graph.*, 31(4):97:1–97:9, July 2012.

[32] C. Zheng and D. L. James. Toward high-quality modal contact sound. *ACM Trans. Graph.*, 30(4):38:1–38:12, July 2011.

A INTERSECTION VOLUME EQUIVALENT SHAPE

The method for estimating the contact patch geometry described in Section 4.2 uses the moments of the intersection volume Ω,

$$V = \int_\Omega 1 \; dxdyxz \tag{14}$$

$$c = \int_\Omega r \; dxdydz \tag{15}$$

$$M = \int_\Omega (r-c)^2 dxdydz. \tag{16}$$

Here r is a point in the volume Ω, V is a scalar giving the volume of the intersection, $c \in \mathbb{R}^3$ is the center, and $M \in \mathbb{R}^{3\times3}$ is the second order moment. Note that when M is diagonalized, the eigenvectors provide an orthogonal set of axes for the second moment tensor, and the eigenvalues are related to an equivalent dimension cuboid of dimensions x, y, and z,

$$I_x = 1/12V(y^2+z^2) \tag{17}$$

$$I_y = 1/12V(x^2+z^2) \tag{18}$$

$$I_z = 1/12V(x^2+y^2). \tag{19}$$

If we desire the equivalent ellipsoid, the constant changes from $1/12$ to $1/5$. Note that when one of the dimensions is very close to zero, suppose the normal direction corresponds to z and it is almost zero as is the case when we have a flat intersection volume, then the moment I_z becomes the sum of the other two.

To compute the x dimension of the equivalent cuboid from the tensor, we use

$$x = \sqrt{\frac{12(I_y + I_z - I_x)}{2V}}, \tag{20}$$

with similar computations for the y and z directions.